Algorithms and computations in physics (Oxford Lectures 2024)

Werner Krauth*
Laboratoire de Physique, Ecole normale supérieure, Paris (France)
Rudolf Peierls Centre for Theoretical Physics & Wadham College
University of Oxford (UK)

Second lecture: 23 January 2024
Third lecture: 30 January 2024
(version: 30/01/2024)

Markov-chain sampling, from the adults' game on the Monte Carlo heliport to modern ideas on non-reversibility.

Contents

2 Markov-chain sampling

We discuss Markov chains, initially in a practical context, but then in (almost) full mathematical rigor. We then walk through an exhibition of algorithms that illustrate some major themes: the

*werner.krauth@ens.fr, werner.krauth@physics.ox.ac.uk

Metropolis algorithm and its modern variants, bounding potentials, non-reversibility, continuous-time Markov processes, event-driven formulations, and thinning. We again close with a discussion of fundamentals: How procisely does a Markov chain "converge" and is it possible, on the Monte Carlo heliport or its discretized variants, to converge completely, so that the outcome cannot be distinguished from that of the children's game?

Background material for this second lecture is contained in the book by Wasserman [1] (basic probability theory and statistics). Levin et al. [2] contains the basic theory of Markov chains, transition matrices, mixing and relaxation times. A recent educational paper with Tartero [3] provides more details on the example algorithms. The heliport game and the Metropolis–Hastings algorithm are treated in more depth in the SMAC book [4] [1]

2.1 Adults on the Monte Carlo heliport

Markov-chain sampling—believe it or not—comes from a game that adults play on the Monte Carlo heliport in the early evenings, with all the helicopters safely stowed away (see Fig. 2.1). Up to a rescaling of lengths, it realizes the same sample space $\Omega^{\square}$ as the children had on the beach.

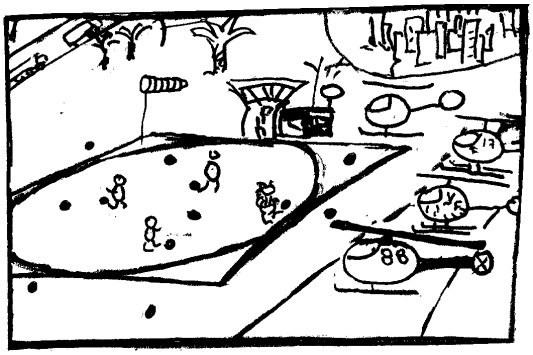

Figure 2.1: Adults throwing pebbles on the Monte Carlo heliport, and computing the number π. The game always starts at the clubhouse. A curious procedure is followed when a pebble falls outside the pad.

Adults fling pebbles inside the square but, because of its sheer size, they cannot possible place pebbles randomly. So they use a different algorithm. They start at the clubhouse (position $(1, 1)$),[2] and carry along with them as many pebbles as their handbags will hold, then throw a pebble randomly around them, go to where it has landed, then throw again, and again. When they throw a pebble outside the square of the heliport, they fetch it and have it placed on top of the pebble that marked their last position, then continue. To clarify the adults' algorithm, we again resort to pseudo-code (see Alg. 2.1 (`markov-pi`)).

Run	N_{hits}	Estimate of π
1	3123	3.123
2	3118	3.118
3	3040	3.040
4	3066	3.066
5	3263	3.263

Table 2.1: Results of five runs of Alg. 2.1 (`markov-pi`) with $N = 4000$ and a throwing range $\delta = 0.3$

[1] NB: "SMAC" = *Statistical Mechanics: Algorithms and Computations* [4]

[2] If they could start at a random position, there would be no point in their game

procedure `markov-pi`
$N_{\rm hits} \leftarrow 0$; $\{x, y\} \leftarrow \{1, 1\}$
for $i = 1, \ldots, N$**:**
  $\Delta_x \leftarrow$ `ran`$(-\delta, \delta)$
  $\Delta_y \leftarrow$ `ran`$(-\delta, \delta)$
  if $|x + \Delta_x| < 1$ and $|y + \Delta_y| < 1$**:** **then**
    $x \leftarrow x + \Delta_x$
    $y \leftarrow y + \Delta_y$
  if $x^2 + y^2 < 1$**:** $N_{\rm hits} \leftarrow N_{\rm hits} + 1$
output $N_{\rm hits}$

Algorithm 2.1: `markov-pi`. Markov-chain Monte Carlo algorithm for computing π in the adults' game. The game starts at the clubhouse; the throwing range δ remains fixed.

At the end of the game, the pattern of pebbles looks weird (see Fig. 2.2), and certainly differs from that in the children's game in Lecture 1. However, when the adults count the number of pebbles inside the circle in the square, they again get a decent—if less precise— approximation of the number π, (see Table 2.1, and again [4, Sect. 1.1] for the full story). We remember in this context that probability theory, as codified in the Kolmogoroff axioms, assigns probabilities in continuous sample spaces not to single samples but to subsets of Ω called *events* (see Ref. [1, Chap. 1.3]). So one can have piles of pebbles yet, per square centimeter, realize a uniform distribution.

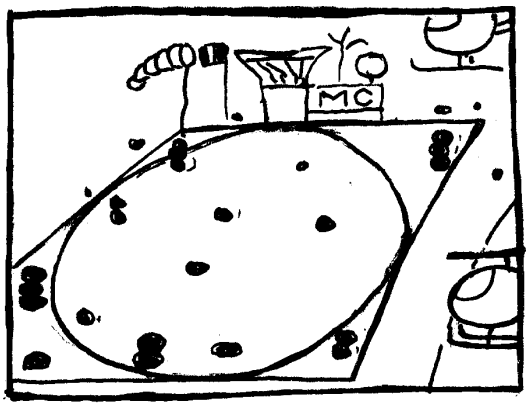

Figure 2.2: Monte Carlo heliport after the game. Piles of pebbles can be seen near the boundaries, and especially near the corners.

The players on the heliport implement the Metropolis algorithm, that dates from 1953 [5]. In this lecture, we will discuss it, prove its correctness, analyze it, then overcome it and confront it with a number of "beyond Metropolis" algorithms.

2.1.1 The transition matrix, balance conditions

To further describe the adults' game and other Markov chains, we imagine its sample space Ω discretized, with a finite number of samples $x \in \Omega$. For concreteness, we suppose the heliport discretized into a 3×3 pebble game with a numbering of configurations from 1 to 9, in the order shown in Fig. 2.3a). The probabilities with which we perform the possible moves, from x to x', then constitute the *transition matrix* $P(x, x')$. In addition, there are time-dependent probability distributions $\pi^{\{t\}}$, where $\pi^{\{t\}}(x)$ describes the probability to be at a position x. At

intermediate times, these distributions $\pi^{\{t\}}$ are insufficiently characterized, but we know that $\pi^{\{t=0\}}$, is a Kronecker δ function on the starting configuration $x_0 = 9$, and that $\pi^{\{t\to\infty\}}$ should be a constant on the landing pad.

The transition matrix P has a double meaning. On the one hand, it encodes a Monte Carlo algorithm: $P(x, x')$ is precisely the (conditional) probability to move from $x_{t-1} = x$ to $x_t = x'$ in one step. The condition $\sum_{x'\in\Omega} P(x, x') = 1$ expresses the conservation of probabilities (where the sum over the x' includes x). On the other hand, the transition matrix gives the relationship between the probability distributions $\pi^{\{t-1\}}$ and $\pi^{\{t\}}$ at subsequent time steps $t-1$ and t:

$$\pi^{\{t\}}(x) = \sum_{x'\in\Omega} \pi^{\{t-1\}}(x')P(x', x) \tag{2.1}$$

(see Fig. 2.3 for an illustration of the double nature of the transition matrix). At time $t = 0$, $\pi^{\{t=0\}}$ is a distribution that we know how to sample. Most of the time, it is a single configuration (in the example, it is a Kronecker δ function on $(x, y) = (3, 3)$.

By extension, the matrix P^t connects the distribution $\pi^{\{t\}}$ with the distribution of initial configurations $\pi^{\{0\}}$ at time $t = 0$:

$$\pi^{\{t\}}(x) = \sum_{x'\in\Omega} \pi^{\{t-1\}}(x')P(x', x) \quad \Rightarrow \quad \pi^{\{t\}}(x) = \sum_{x'\in\Omega} \pi^{\{0\}}(x')(P^t)(x', x) \quad \forall\sigma \in \Omega, \tag{2.2}$$

where we pay attention to the fact that the matrix P^2, for example, is defined as the matrix product:

$$(P^2)(x, x') = \sum_{x''\in\Omega} P(x, x'')P(x'', x') \quad \forall x, x' \in \Omega \tag{2.3}$$

and likewise for higher powers of t. The matrix $P^2(x, x')$ describes to probability to move from x to x' in *two* steps, etc.

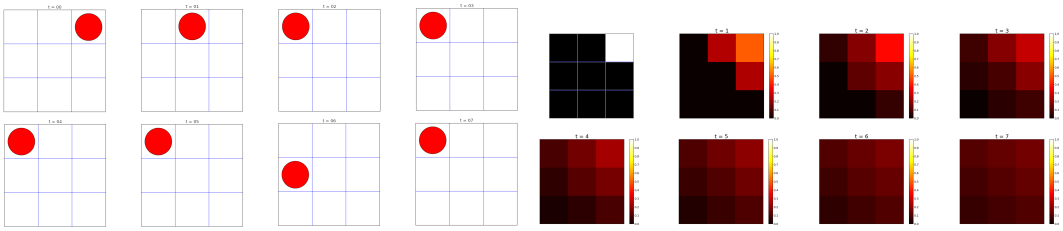

Figure 2.3: Double role of transition matrix in the 3×3 pebble game. (a): P encodes the Monte Carlo algorithm. (b): P governs the evolution of the probability $\pi^{\{t\}}$.

We want the distribution $\pi^{\{t\}}$ at later times to correspond to the distribution π (in the example of the heliport and its 3×3 discretization, we want it to be be uniform $\pi^{\{t\to\infty\}} = \frac{1}{9}$). Dropping the time indices in eq. (2.1), we arrive at the *global-balance condition*

$$\pi(x) = \sum_{x'\in\Omega} \pi(x')P(x', x) \quad \forall x \in \Omega. \tag{2.4}$$

which is a condition on the transition matrix P (in other words, the Monte Carlo algorithm) for an imposed distribution π that we want to sample[3] For a transition matrix that is *irreducible*, the global-balance condition is satisfied for a unique stationary distribution π. "Irreducible" means that (for a finite Ω) the probability to move in a finite time from any x to any x' is finite.

Any irreducible transition matrix has a unique π, but this distribution is not necessarily the limit $\pi^{\{t\}}$ for $t \to \infty$ for all initial distributions $\pi^{\{0\}}$. A simple illustration of this consists, in

[3] In non-equilibrium statistical mechanics, one is often interested in the reverse problem: determining the steady state π for a given transition matrix P.

the 3×3 pebble game, in a transition matrix $P(1,2) = 1$, $P(2,3) = 1$ to $P(9,1) = 1$, with some numbering of the nine configurations. Convergence towards π of an irreducible Markov chain requires that it is aperiodic, that is, that the return times from a configuration i back to itself $\{t \geq 1 : (P^t)(x,x) > 0\}$ are not all multiples of a period larger than one. For irreducible, aperiodic transition matrices, $P^t = (P^t)(x,x')$ is a positive matrix for some fixed t, and MCMC converges towards π from any starting distribution $\pi^{\{0\}}$.

In conclusion, Markov chains that satisfy a single minimal—and easily verifiable—requirement (irreducibility) have a unique stationary distribution π. Under a second equally simple requirement (aperiodicity), they satisfy $\pi^{\{t\}} \to \pi$ for any initial distribution $\pi^{\{t=0\}}$. We will argue time and again that one has to wait until $\pi^{\{t\}}$ is close to π before extracting useful information from a Monte Carlo calculation.

2.1.2 Reversible and non-reversible Markov chains

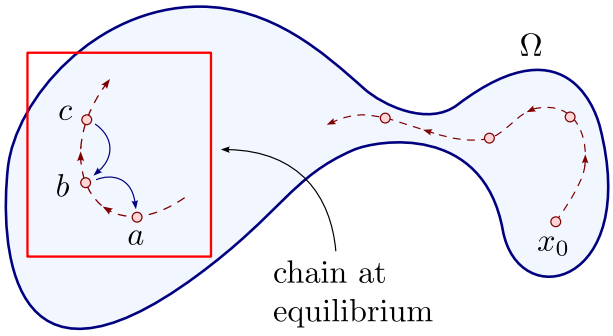

Figure 2.4: Motion of a Markov chain in equilibrium. For a reversible Markov chain in equilibrium, the trajectory $a \to b \to c$ appears with the same probability as the "reverse" trajectory $c \to b \to a$.

Reversible algorithms are those that satisfy the *detailed-balance condition*

$$\pi(x)P(x,x') = \pi(x')P(x',x) \quad \forall x, y \in \Omega. \tag{2.5}$$

Detailed balance implies global balance (eq. (2.5) yields eq. (2.4) by summing over x', considering that $\sum_{x' \in \Omega} P(x,x') = 1$). The detailed-balance condition imposes that in equilibrium, the path from x (at time $t-1$) to x' (at time t) is equally likely as the time-reversed path from x' to x. This can be extended to paths that are arbitrary long (see Fig. 2.4), and explains why Markov chains that satisfy the detailed-balance condition are equivalently called *reversible*.

To set up a reversible transition matrix P for a given distribution π, we may choose

$$\pi(x)P(x,x') \propto \min\left[\pi(x), \pi(x')\right] \quad \text{for } x \neq x'. \tag{2.6}$$

The right-hand side of Eq. (2.6) is symmetric in x and x', so that the left-hand side must also be symmetric. Therefore, detailed balance is automatically satisfied. We divide both sides by $\pi(x)$ and arrive at the equation famously proposed by Metropolis et al. in 1953:

$$P^{\text{Met}}(x,x') \propto \min\left[1, \frac{\pi(x')}{\pi(x)}\right] \quad \text{for } x \neq x'. \tag{2.7}$$

Let us discuss the difference between a transition matrix and a filter in order to make Eq. (2.7) explicit and remove the proportionality sign. The move from x to $x' \neq x$ proceeds in two steps. A possible move is first proposed with an *a priori* probability $\mathcal{A}(x,x')$ and is then accepted or rejected with a filter. In the Metropolis algorithm, the *a priori* probability is symmetric,

$\mathcal{A}(x,x') = \mathcal{A}(x',x)$, and

$$\underbrace{P^{\rm Met}(x,x')}_{\text{transition matrix}} = \underbrace{\mathcal{A}(x,x')}_{\textit{a priori}\text{ probability}} \overbrace{\mathcal{P}^{\rm Met}(x,x')}^{\text{Metropolis filter}} . \tag{2.8}$$

For the Metropolis algorithm, a proposed move $x \to x'$ (with $x' \neq x$) is thus accepted with the Metropolis acceptance probability

$$\mathcal{P}^{\rm Met}(x,x') = \min\left[1, \frac{\pi(x')}{\pi(x)}\right], \tag{2.9}$$

which is also called the *Metropolis filter* in order to differentiate it from the transition matrix. If the move $x \to x'$ is rejected, the particle remains at x, which determines the diagonal transition matrix elements $P(x,x)$ and guarantees that $\sum_{x'} P(x,x') = 1$, in other words, that the transition matrix is *stochastic*.

For the transition matrix P of a reversible Markov chain, the matrix $A_{ij} = \pi_i^{1/2} P_{ij} \pi_j^{-1/2}$ is symmetric, as trivially follows from the detailed balance of eq. (2.5). The spectral theorem then assures that A has only real eigenvalues and that its eigenvectors form an orthonormal basis. The transition matrix P has the same eigenvalues as A, as well as closely related (right) eigenvectors:

$$\sum_{j\in\Omega} \underbrace{\pi_i^{1/2} P_{ij} \pi_j^{-1/2}}_{A_{ij}} x_j = \lambda x_i \quad \Leftrightarrow \quad \sum_{j\in\Omega} P_{ij} \underbrace{\left[\pi_j^{-1/2} x_j\right]}_{\tilde{x}_j} = \lambda \underbrace{\left[\pi_i^{-1/2} x_i\right]}_{\tilde{x}_i} . \tag{2.10}$$

The eigenvectors $\tilde{x}$ of P must be multiplied with $\sqrt{\pi}$ to be mutually orthogonal. They provide a basis on which any initial probability distribution $\pi^{\{0\}}$ can be expanded. An irreducible and aperiodic transition matrix P (reversible or not) has one eigenvalue $\lambda_1 = 1$, and all others satisfy $|\lambda_k| < 1 \; \forall k \neq 1$. The unit eigenvalue λ_1 corresponds to a constant right eigenvector of P because of the stochasticity condition $\sum_{j\in\Omega} P_{ij} = 1$, and to the left eigenvector π of P, because of the global-balance condition of eq. (2.4). Let us consider a reversible transition matrix with a non-degenerate spectrum (which must be real, as we just showed), then slightly perturb the elements of P, which will make it non-reversible. As the eigenvalues of a matrix continuously depends on its elements, it follows that a non-reversible transition matrix may very well have a real-valued spectrum.

2.1.3 Metropolis–Hastings algorithm

In Alg. 2.1 (`markov-pi`), moves (Δ_x, Δ_y) are restricted to a small square of edge length 2δ, the throwing range, and as this throwing range around a position (x,y) is independent of the position, the a priori probability is symmetric, and even constant (see Fig. 2.5*A*). The small square could be replaced by a small disk without bringing in anything new (see Fig. 2.5*B*). A more interesting situation arises for asymmetric a priori probabilities: in the triangle algorithm of Fig. 2.5*C*, moves are sampled from an oriented equilateral triangle centered at a, with one edge parallel to the x-axis. This extravagant choice may lack motivation in the context of the adults' game, but contains a crucial ingredient of modern Monte Carlo algorithms, that we will study in later lectures.

The detailed-balance condition of eq. (2.5), in the presence of an asymmetric a priori probability $\mathcal{A}(x,x')$, such as the triangular one, gives

$$\pi(x)\mathcal{A}(x,x')\mathcal{P}(x,x') = \pi(x')\mathcal{A}(x',x)\mathcal{P}(x',x'). \tag{2.11}$$

In this equation, π is given (we want it to be uniform, on the heliport) and so is $\mathcal{A}$ (we want it

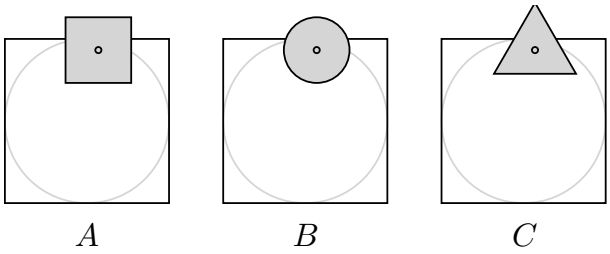

Figure 2.5: Throwing pattern in Alg. 2.1 (`markov-pi`) (A), with variants. The triangle algorithm (C) requires special attention.

to be uniform, in an equilateral triangle). The probability of moving from a to b must satisfy $\pi(a)\mathcal{P}(a \to b) = \pi(b)\mathcal{P}(b \to a)$, so that the acceptance probabilities (the filter) must obey:

$$\frac{\mathcal{P}^{\text{Met–H}}(x, x')}{\mathcal{P}^{\text{Met–H}}(x', x)} = \frac{\pi(x')}{\mathcal{A}(x, x')} \frac{\mathcal{A}(x', x)}{\pi(x)}.$$

leading to

$$\mathcal{P}^{\text{Met–H}}(x, x') = \min \left[1, \frac{\pi(x')}{\mathcal{A}(x, x')} \frac{\mathcal{A}(x', x)}{\pi(x)} \right], \tag{2.12}$$

also called the Metropolis–Hastings filter, which always requires one to take into consideration the back move (from x' to x), when one decides whether one should accept the move from x to x'.

2.2 Reversible Markov chains (picture book)

The material in this section is taken from Ref. [3], which presents a dozen of distinct Markov-chain Monte Carlo algorithms to sample the Boltzmann distribution of the anharmonic oscillator. We present three reversible Markov chains (plus a patch) before continuing with four non-reversible algorithms, in Sec. 2.3.

2.2.1 Metropolis algorithm

Algorithm 2.2 (`metropolis`) implements the symmetric *a priori* probability as a uniform displacement $\Delta = x' - x$ which is as likely as $-\Delta$. The Metropolis filter is implemented with a uniform random number Υ between 0 and 1, which we refer to as a "pebble." For large times t, when the initial configuration is forgotten, the algorithm samples π_{24}. In all the following Markov-chain algorithms, this large t condition is understood.

procedure `metropolis`
input x (sample at time t)
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min \left[1, \dfrac{\pi_{24}(x')}{\pi_{24}(x)} \right]$ **:** $x \leftarrow x'$
output x (sample at time $t+1$)

Algorithm 2.2: `metropolis`. Sampling π_{24} with the Metropolis algorithm.

2.2.2 Factorized Metropolis algorithm

The Metropolis algorithm is really famous, but it is not the end of history. A modern variant is the factorized Metropolis algorithm that we do not discuss here in detail, but only apply to the anharmonic oscillator, where it is written as:

$$\mathcal{P}_{24}^{\text{fact}}(x, x') = \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right] \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]. \tag{2.13}$$

The factorized Metropolis algorithm satisfies detailed balance:

$$\begin{aligned} &\pi_{24}(x) P_{24}^{\text{fact}}(x, x') \\ &\propto \underbrace{\pi_2(x) \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right]}_{\min[\pi_2(x), \pi_2(x')]:\ x \Leftrightarrow x'} \underbrace{\pi_4(x) \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]}_{\min[\pi_4(x), \pi_4(x')]:\ x \Leftrightarrow x'} \\ &\propto \pi_{24}(x') P_{24}^{\text{fact}}(x', x), \end{aligned} \tag{2.14}$$

where we have dropped the symmetric *a priori* probability $\mathcal{A}$. Algorithm 2.3 (`factor-metropolis`) implements the factorized Metropolis filter: Algorithm 2.3 (`factor-metropolis`) (which is

procedure `factor-metropolis`
input x
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min\left[1, \frac{\pi_2(x')}{\pi_2(x)}\right] \min\left[1, \frac{\pi_4(x')}{\pi_4(x)}\right]$ **:**
$\{\ x \leftarrow x'$
output x

Algorithm 2.3: `factor-metropolis`. Sampling π_{24} naively with the factorized Metropolis filter (see Ref. [3]).

naive) can be patched by replacing its random number Υ by two independent random numbers Υ_2 and Υ_4, as shown in Alg. 2.4 (`factor-metropolis(patch)`). There, a proposed move is accepted by *consensus* if all the factors accept it. In Alg. 2.4 (`factor-metropolis(patch)`), two independent decisions are taken,[4] one for the harmonic and one for the quartic factor, and the proposed move is finally accepted only if it is accepted by both factors. The output is identical to that of Alg. 2.3 (`factor-metropolis`).

2.2.3 Bounding potentials

Monte Carlo algorithms (at a difference with molecular-dynamics algorithms that we will discuss in later lectures) are decision problems where proposed moves are accepted with a filter, for example the Metropolis filter $\min[1, \exp(-\beta\Delta U)]$. One can often base the accept/reject decision on a bounding potential $\widehat{U}$, and thus avoid computing U, ΔU, and their exponentials

We say that $\widehat{U}$ is a bounding potential of a potential U if, for any pair of configurations x and x', it satisfies

$$\min\left(1, \mathrm{e}^{-\beta\Delta\widehat{U}}\right) \leq \min\left(1, \mathrm{e}^{-\beta\Delta U}\right) \quad \forall\, x, x' \in \Omega, \tag{2.15}$$

[4] one can view this as the sampling of two independent Boolean random variables, (see Ref. [6]), of which the final decision is the *conjunction*.

procedure `factor-metropolis(patch)`
input x
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$
$\Upsilon_2 \leftarrow \texttt{ran}(0,1)\,;\ \Upsilon_4 \leftarrow \texttt{ran}(0,1)$
if $\Upsilon_2 < \min\left[1, \dfrac{\pi_2(x')}{\pi_2(x)}\right]$ and $\Upsilon_4 < \min\left[1, \dfrac{\pi_4(x')}{\pi_4(x)}\right]$ **:**
$\{\ x \leftarrow x'$ (move accepted by consensus)
output x

Algorithm 2.4: `factor-metropolis(patch)`. Patch of Alg. 2.3, implementing the consensus principle (see Ref. [3]).

where $\Delta\widehat{U} = \widehat{U}(x') - \widehat{U}(x)$ and $\Delta U = U(x') - U(x)$. [5] Concretely, we define harmonic and quartic bounding potentials recursively for $n = 0, \pm 1, \pm 2$, etc., as

$$\widehat{U}_2(n) = \begin{cases} 0 & \text{if } n = 0 \\ \widehat{U}_2(|n|-1) + |n| & \text{if } n \in \mathbb{Z}\backslash\{0\} \end{cases}, \qquad \widehat{U}_4(n) = \begin{cases} 0 & \text{if } n = 0 \\ \widehat{U}_4(|n|-1) + |n^3| & \text{if } n \in \mathbb{Z}\backslash\{0\}. \end{cases} \tag{2.16}$$

These definitions are extended to non-integer arguments x by linear interpolation. The anharmonic bounding potential is then defined as $\widehat{U}_{24}(x) = \widehat{U}_2(x) + \widehat{U}_4(x)$ (see Fig. 2.6).

A bounding potential can simplify the decision to accept a move because a pebble $0 < \Upsilon < 1$ that falls below $\exp\bigl(-\beta\Delta\widehat{U}\bigr)$ also falls below $\exp\bigl(-\beta\Delta U\bigr)$ (see Fig. 2.7a). In the remaining algorithms, we use a two-pebble strategy for the decision to accept or reject a move. The first pebble $0 < \Upsilon_1 < 1$ decides whether a move is accepted with respect to the bounding potential. Otherwise, if Υ_1 rejects the move, we use a second pebble Υ_2 to decide whether the first-pebble rejection with respect to $\widehat{U}$ stands with respect to U (see Fig. 2.7b). A rescaling, with $0 < \Upsilon_2 < 1$, allows us to definitely reject the move if

$$\Upsilon_2 < \frac{1 - \mathrm{e}^{-\beta\Delta U}}{1 - \mathrm{e}^{-\beta\Delta\widehat{U}}}. \tag{2.17}$$

The two-pebble bounding-potential algorithm is implemented in Alg. 2.5 (`bounded-metropolis`) for the anharmonic oscillator. It again samples the Boltzmann distribution π_{24}, although most positive decisions are taken on the basis of the bounding potential.

2.3 Non-reversible Markov chains (picture book)

In a tradition that started with the Metropolis algorithm many decades ago, Markov chains are normally designed with the restrictive detailed-balance condition, although they are only required to satisfy global balance. In this section, we illustrate more recent attempts to overcome the detailed-balance condition in a systematic way, within the framework of "lifted" Markov chains. Background and references can be found in [3].

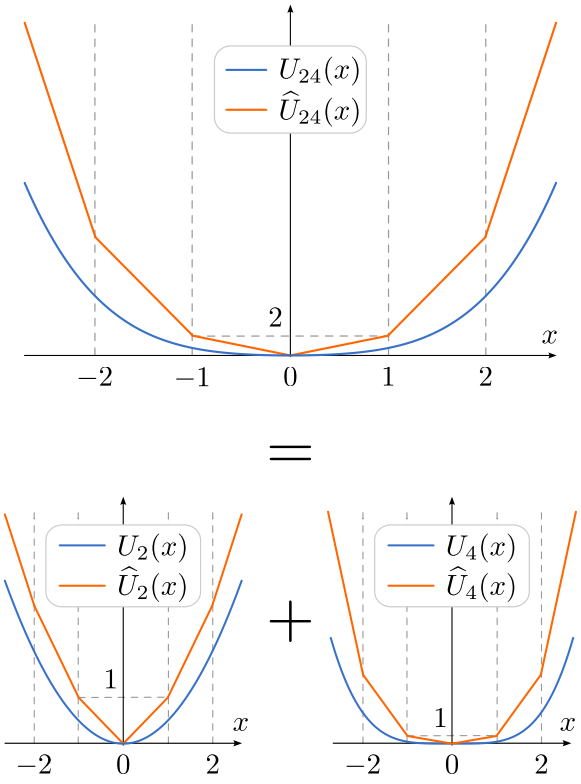

Figure 2.6: Anharmonic bounding potential $\widehat{U}_{24}$ and its harmonic and quartic constituents $\widehat{U}_2$ and $\widehat{U}_4$ (see Ref. [3]).

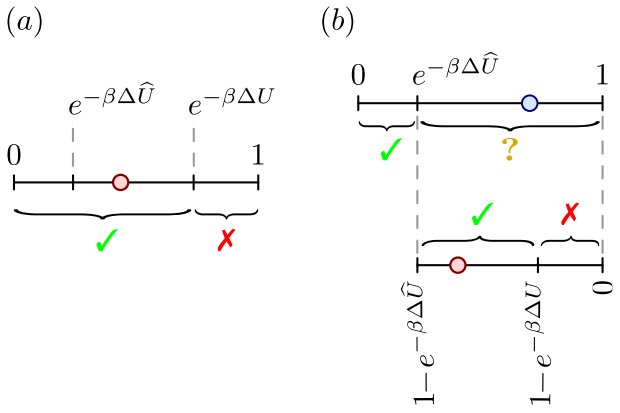

Figure 2.7: Single-pebble and two-pebble decisions in the Metropolis algorithm. (a): A single pebble Υ illustrating that acceptance with respect to the bounding potential implies acceptance with respect to U. (b): The first pebble Υ_1 makes a decision with respect to the bounding potential. In case of rejection, a second pebble Υ_2 definitely decides on the move (see Ref. [3]).

procedure `bounded-metropolis`
input x
$\Delta \leftarrow \texttt{ran}(-\delta, \delta)$
$x' \leftarrow x + \Delta$; $\Upsilon_1 \leftarrow \texttt{ran}(0,1)$;
if $\Upsilon_1 < \min\left(1, \mathrm{e}^{-\beta\Delta\widehat{U}_{24}}\right)$ **:**
$\{\ x \leftarrow x'$
else:
$\Upsilon_2 \leftarrow \texttt{ran}(0,1)$
if $\Upsilon_2 > \dfrac{1-\mathrm{e}^{-\beta\Delta U_{24}}}{1-\mathrm{e}^{-\beta\Delta\widehat{U}_{24}}}$**:** $x \leftarrow x'$
output x

Algorithm 2.5: `bounded-metropolis`. Metropolis algorithm illustrating the use of bounding potentials with a two-pebble decisions. The second pebble is used and the true potential U_{24} is evaluated only after a first-pebble rejection with respect to the bounding potential $\widehat{U}_{24}$.

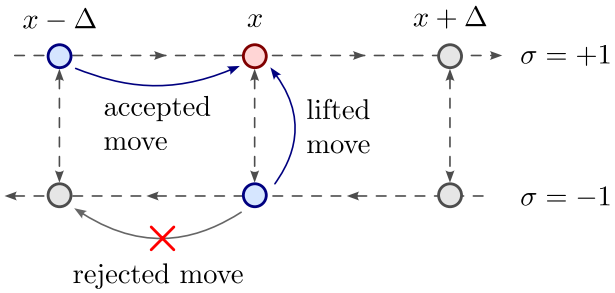

Figure 2.8: Discretized lifted Metropolis algorithm for the anharmonic oscillator. The flow into the lifted configuration $\{x, +1\}$ is indicated [see Eq. (2.21)].

2.3.1 Lifting and the zig-zag algorithm

The Metropolis algorithm proposes positive and negative displacements Δ for the anharmonic oscillator with symmetric *a priori* probabilities (see Alg. 2.2 (`metropolis`)). The filter then imposes that the net flow vanishes, so there will be as many particles going from x to $x+\Delta$ as in the reverse direction, even if, say, $\pi(x) \ll \pi(x+\Delta)$. To break detailed balance and only satisfy global balance, (while keeping π_{24} as a stationary distribution), we first suppose that the positions x lie on the grid $\{\ldots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \ldots\}$, with moves allowed only between nearest neighbors. Each configuration x is duplicated into a forward-moving one $\{x, +1\}$, and a backward-moving one $\{x, -1\}$. From a lifted configuration $\{x, \sigma\}$, the lifted Metropolis algorithm proposes only a forward move if $\sigma = 1$, and only a backward move if $\sigma = -1$. In summary,

$$P^{\text{lift}}(\{x,\sigma\},\{x+\sigma\Delta,\sigma\}) = \min\left[1, \frac{\pi_{24}(x+\sigma\Delta)}{\pi_{24}(x)}\right], \tag{2.18}$$

where $\sigma = \pm 1$. When this move is rejected by the Metropolis filter, the algorithm flips the direction and instead moves from $\{x, \sigma\}$ to $\{x, -\sigma\}$:

$$P^{\text{lift}}(\{x,\sigma\},\{x,-\sigma\}) = 1 - \min\left[1, \frac{\pi_{24}(x+\sigma\Delta)}{\pi_{24}(x)}\right]. \tag{2.19}$$

[5] Equation (2.15) is more restrictive than necessary in the general case.

This algorithm clearly violates detailed balance as there is thus no backward flow for $\sigma = +1$ and no forward flow for $\sigma = -1$. On the other hand, the lifted Metropolis algorithm satisfies the global-balance condition of Eq. (2.4) with the ansatz

$$\pi_{24}^{\text{lift}}(\{x, \sigma\}) = \frac{1}{2}\pi_{24}(x) \quad \text{for } \sigma = \pm 1. \tag{2.20}$$

For example, the flow into the lifted configuration $\{x, +1\}$ satisfies

$$\begin{aligned} &\pi_{24}(\{x, +1\}) \\ &= \pi_{24}(\{x - \Delta, +1\}) P^{\text{lift}}(\{x - \Delta, +1\}, \{x, +1\}) \\ &+ \pi_{24}(\{x, -1\}) P^{\text{lift}}(\{x, -1\}, \{x, +1\}). \end{aligned} \tag{2.21}$$

The two contributions on the right-hand side of Eq. (2.21) correspond on the one hand to the accepted moves from $\{x-\Delta, +1\}$, and on the other hand to the lifted moves from $\{x, -1\}$, when the move from $\{x, -1\}$ toward $\{x - \Delta, -1\}$ is rejected (see Fig. 2.8). Equation (2.21) can be transformed into

$$\begin{aligned} \pi_{24}(x) &= \pi_{24}(x - \Delta) \min\left[1, \frac{\pi_{24}(x)}{\pi_{24}(x - \Delta)}\right] \\ &+ \pi_{24}(x) \left\{1 - \min\left[1, \frac{\pi_{24}(x - \Delta\sigma)}{\pi_{24}(x)}\right]\right\}, \end{aligned} \tag{2.22}$$

which is identically satisfied. We have shown that the lifted Metropolis algorithm satisfies the global-balance condition for the ansatz of Eq. (2.20), which splits $\pi_{24}(x)$ equally between $\{x, +1\}$ and $\{x, -1\}$. The sequence $\pi^{\{t\}}$ will actually converge to this stationary distribution.

In the lifted Metropolis algorithm, the particle, starting from $x_0 = 0$, climbs uphill in direction σ until a move is rejected by the filter, when it remains at its current position but reverses its velocity to $-\sigma$. The following downhill moves, again without rejections, are followed by another uphill climb, and so on, criss-crossing between the two wings of the potential U_{24}. It outputs configurations $\{x, \sigma\}$ such that, remarkably, the x-component samples π_{24}. This curious algorithm is implemented in Alg. 2.6 (`lifted-metropolis`).

procedure `lifted-metropolis`
input $\{x, \sigma\}$ (lifted sample at time t)
$\Delta \leftarrow \texttt{ran}(0, \delta)$ $(\delta > 0)$
$x' \leftarrow x + \sigma\Delta$ (x' in direction σ from x)
$\Upsilon \leftarrow \texttt{ran}(0, 1)$
if $\Upsilon < \min\left[1, \dfrac{\pi_{24}(x')}{\pi_{24}(x)}\right]$ **:** $x \leftarrow x'$
else: $\sigma \leftarrow -\sigma$
output $\{x, \sigma\}$ (lifted sample at time $t + 1$)

Algorithm 2.6: `lifted-metropolis`. Non-reversible lifted version of Alg. 2.2 (`metropolis`). The x-positions that are output by this program sample π_{24} (see Ref. [3]).

2.3.2 Event-driven Markov processes

Markov chains in continuous time are *Markov processes*. Algorithm 2.6 (`lifted-metropolis`) with its grid of positions $\{\dots, -2\Delta, -\Delta, 0, \Delta, 2\Delta, \dots\}$ and nearest-neighbor moves can be studied in the limit of very small Δ, where one may rescale time such that a displacement $\pm\Delta$ is itself undertaken in a time interval Δ. The particle in the anharmonic oscillator thus moves with

unit absolute velocity, whose sense is reversed when there is a rejection. The downhill moves are all accepted, and even uphill moves are accepted with a probability close to one. Rather than to simulate each of these steps, we sample the position of the next rejection. As an example, let us consider a sequence of uphill moves in positive direction from $x = 0$. The probability for accepting an entire sequence of n subsequent uphill moves, at positions $0, \Delta, \ldots, (n-1)\Delta$, and then rejecting the move $n+1$, is

$$\mathbb{P}(0 \to x_{\rm ev}) = \underbrace{\mathrm{e}^{-\beta\Delta U_{24}(0)\cdots\Delta U_{24}[(n-1)\Delta]}}_{n \text{ accepted moves}} \underbrace{\left[1 - \mathrm{e}^{-\beta\Delta U_{24}(n\Delta)}\right]}_{\text{rejection}} \to \beta \mathrm{e}^{-\beta U_{24}} \mathrm{d}U_{24}. \tag{2.23}$$

In the small-Δ limit, the rejection is here expanded to first order, and ΔU is replaced by $\mathrm{d}U$. In our example of the anharmonic oscillator starting at $x = 0$, all the increments of ΔU_{24} up to position x add up to the potential $U_{24}(x)$. Equation (2.23) indicates that the value of U_{24} at which the velocity is reversed follows an exponential distribution in U_{24}. Remembering from Lecture 1 how to sample an exponential random variable, we obtain

$$U_{24}(x_{\rm ev}) = -\beta^{-1} \log \texttt{ran}(0,1)\,, \tag{2.24}$$

which can be inverted as $U_{24}(x_{\rm ev}) = x_{\rm ev}^2/2 + x_{\rm ev}^4/4$, with

$$x_{\rm ev} = \sigma\sqrt{-1 + \sqrt{1 - 4\beta^{-1} \log \texttt{ran}(0,1)}}. \tag{2.25}$$

Algorithm 2.7 (`zig-zag`) implements this event-driven, continuous-time, Markov process and manages to move forward and backward. The equal-time samples again sample the Boltzmann distribution π_{24} (see Fig. 2.9). The algorithm was, in essence, proposed in 2012 [7].

procedure `zig-zag`
input $\{x, \sigma\}, t$ (lifted sample with $\sigma x \le 0$)
$x_{\rm ev} \leftarrow \sigma\sqrt{-1 + \sqrt{1 - 4\beta^{-1} \log \texttt{ran}(0,1)}}$ (see Eq. (2.25))
$t_{\rm ev} \leftarrow t + |x_{\rm ev} - x|$
for $t^* = \mathrm{int}(t) + 1, \ldots, \mathrm{int}(t_{\rm ev})$**:**
  { **print** $x + \sigma(t^* - t)$ (equal-time samples)
$x \leftarrow x_{\rm ev};\ \sigma \leftarrow -\sigma;\ t \leftarrow t_{\rm ev}$ ("zig-zag")
output $\{x, \sigma\}, t$

Algorithm 2.7: `zig-zag`. Continuous-time, event-driven version of Alg. 2.6 (`lifted-metropolis`). The x-positions output by the `print` statement sample π_{24} (see Ref. [3]).

2.3.3 Thinning and the avoidance of evaluation

The Algorithm 2.7 (`zig-zag`) is the tip of the iceberg of modern non-reversible Markov chains, but let us go even farther in our exploration of miniature programs, for example by considering Alg. 2.8 (`bounded-lifted`), which differs by a single line from Alg. 2.5 (`bounded-metropolis`), and is self-explanatory. Let us finally take to continuous-time limit of this algorithm, where the particle moves up the bounding potential $\widehat{U}_{24}$, rather that the true potential U_{24} (see Fig. 2.10). When it must throw the second pebble Υ_2, to find out whether the rejection is real, it uses:

$$\Upsilon_2 < \frac{1 - \mathrm{e}^{-\beta\Delta U_{24}}}{1 - \mathrm{e}^{-\beta\Delta \widehat{U}_{24}}} \to \Upsilon_2 < \frac{\mathrm{d}U_{24}/\mathrm{d}x}{\mathrm{d}\widehat{U}_{24}/\mathrm{d}x}, \tag{2.26}$$

as implemented in Alg. 2.9 (`bounded-zig-zag`), a truly remarkable 11-line algorithm.

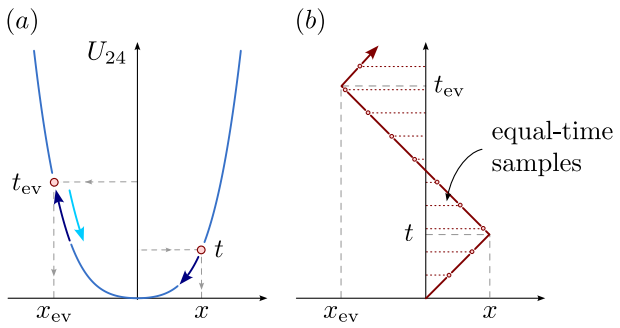

Figure 2.9: Zig-zag algorithm (continuous-time event-driven lifted Metropolis chain). (a): The particle swings about the origin, turning around at positions $x_{\rm ev}$ [sampled by Eq. (2.25)]. (b): Piecewise deterministic constant-velocity trajectory. Particle positions are sampled at equal time steps (see Ref. [3]).

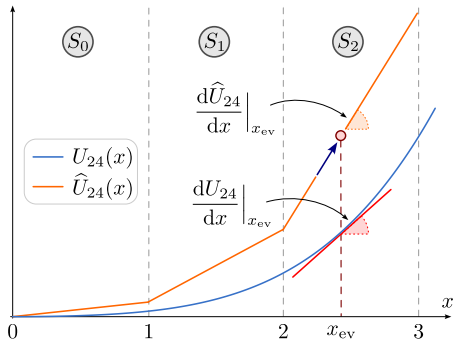

Figure 2.10: Continuous-time version of the bounded-lifted Metropolis algorithm as implemented in Alg. 2.9 (`bounded-zig-zag`) The proposed event $x_{\rm ev}$ is confirmed by comparing the derivatives of the true potential U_{24} and the bounding potential $\widehat{U}_{24}$ [see Eq. (2.26)].

procedure `bounded-lifted`
input $\{x, \sigma\}$ (lifted sample at time t)
$\Delta \leftarrow \texttt{ran}(0, \delta)$ $(\delta > 0)$
$x' \leftarrow x + \sigma\,\Delta$; $\Upsilon_1 \leftarrow \texttt{ran}(0,1)$
if $\Upsilon_1 < \min\left(1, \mathrm{e}^{-\beta\Delta\widehat{U}_{24}}\right)$**:**
$\{\ x \leftarrow x'$
else:
$\Upsilon_2 \leftarrow \texttt{ran}(0,1)$
if $\Upsilon_2 > \dfrac{1-\mathrm{e}^{-\beta\Delta U_{24}}}{1-\mathrm{e}^{-\beta\Delta\widehat{U}_{24}}}$**:** $x \leftarrow x'$
else: $\sigma \leftarrow -\sigma$
output $\{x, \sigma\}$ (lifted sample at time $t+1$)

Algorithm 2.8: `bounded-lifted`. Discrete-time bounded-lifted Metropolis algorithm using two-pebble decisions. The second pebble is used and the true potential U_{24} is evaluated only after a first-pebble rejection with respect to the bounding potential $\widehat{U}_{24}$.

2.4 Convergence of Markov chains: fundamental aspects

In the previous sections of this lecture, we established the correctness of a number of reversible and non-reversible algorithms. Implementing and running the associated computer programs, we convinced ourselves that the algorithms where correct and that, for $t \to \infty$, the distribution $\pi^{\{t\}}$ converged towards $\pi^{\{t=\infty\}} = \pi$. On the heliport, π corresponds to the uniform distribution in the square. In the anharmonic oscillator, it corresponds to the Boltzmann distribution $\exp(-\beta U_{24})$ with $U_{24}(x) = x^2/2 + x^4/4$.

The convergence of the $\pi^{\{t\}}$ to π differs in nature from the convergence of the running average (the number of hits to trials in the pebble games of the last two lectures).

In this context, one defines the *total-variation distance* (TVD) of two (normalized) distributions ρ and π by

$$||\rho - \pi||_{\mathrm{TV}} = \max_{A\subset\Omega} |\rho(A) - \pi(A)| = \frac{1}{2}\sum_{x\in\Omega} |\rho(x) - \pi(x)| \tag{2.27}$$

2.4.1 Convergence theorem of Markov chain, ergodic theorem

Convergence theorem for Markov chains: Suppose that the transition matrix P of a finite Markov chain is irreducible and aperiodic, with stationary distribution π. Then there exist constant $\alpha \in (0,1)$ and $C > 0$ such that

$$\max_{x_0\in\Omega} ||P^t(x_0, .) - \pi||_{\mathrm{TV}} \le C\alpha^t \tag{2.28}$$

(see [2, Theorem 4.9]. The proof is just a few lines long.). Exponential convergence can thus be proven for all initial conditions, with minimal requirements on the Markov chain.

Ergodic theorem for Markov chains: Suppose that the transition matrix P of a finite Markov chain is irreducible, with stationary distribution π. Consider a real-valued function f, and a starting distribution μ that can be arbitrarily different from π. Then

$$\mathbb{P}_\mu\left\{\lim_{t\to\infty}\frac{1}{t}\sum_{s=1}^{t} f(x_s) = \langle f\rangle\right\} = 1 \tag{2.29}$$

generalizing the strong law of large numbers.

procedure `bounded-zig-zag`
input $\{x, \sigma\}, t$ (lifted sample)
if $\sigma x < 0$**:** $x_0 \leftarrow 0$; **else:** $x_0 \leftarrow x$ (starting point)
$n \leftarrow \text{int}(|x_0|)$; $\hat{q} \leftarrow n + 1 + (n+1)^3$; $\tilde{\sigma} \leftarrow \sigma$
$x_{\text{ev}} \leftarrow x_0 + \sigma \left[-(\beta\hat{q})^{-1} \log \texttt{ran}(0,1)\right]$
if $|x_{\text{ev}}| > n + 1$**:**
  $\{$ $x_{\text{ev}} \leftarrow \sigma(n+1)$
else if $\texttt{ran}(0,1) < |x_{\text{ev}} + x_{\text{ev}}^3|/\hat{q}$**:**
  $\{$ $\tilde{\sigma} \leftarrow -\sigma$
$t_{\text{ev}} \leftarrow t + |x_{\text{ev}} - x|$
for $t^* = \text{int}(t) + 1, \ldots, \text{int}(t_{\text{ev}})$**:**
  $\{$ **print** $x + \sigma(t^* - t)$ (equal-time samples)
$x \leftarrow x_{\text{ev}}$; $\sigma \leftarrow \tilde{\sigma}$; $t \leftarrow t_{\text{ev}}$ ("zig-zag")
output $\{x, \sigma\}, t$

Algorithm 2.9: `bounded-zig-zag`. Continuous-time bounded-lifted Metropolis algorithm. It need not invert the potential U_{24}, foreshadowing the use of bounding potentials in real-world applications.

2.4.2 Stopping times

After having concentrated on the mere subject of their correctness, we now discuss the detailed behavior of Markov chains, and introduce the top-to-random shuffling of cards, which has revolutionized the understanding of convergence. As its name indicates, the algorithm consists in taking the top card of the deck, and re-inserting it randomly. We use a formulation of the algorithm where, with n cards, we have n equally likely places where to put it, including where it was before (see Fig. 2.11).

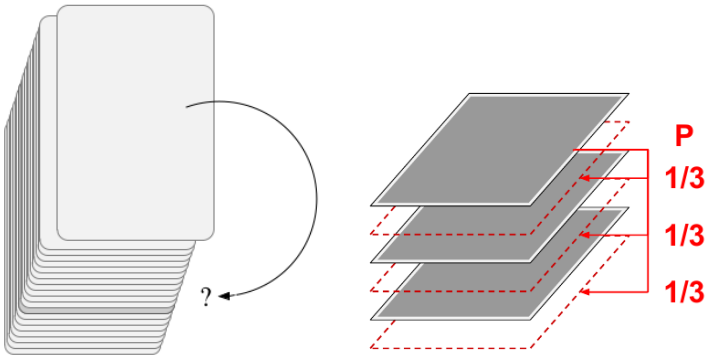

Figure 2.11: Top-to-random shuffle. (a): General presentation. (b): The n choices of placing the top card, for $n = 3$ (see Alg. 2.10 (`top-to-random`).)

The sample space $\Omega_n^{\text{shuffle}}$ of the top-to-random shuffle of n cards consists of all the permutations of $\{1, \ldots, n\}$, so that, for $n = 3$, we have

$$\Omega_3^{\text{shuffle}} = \{1 \equiv \{1,2,3\}, 2 \equiv \{1,3,2\}, 3 \equiv \{2,1,3\}, \tag{2.30}$$
$$4 \equiv \{2,3,1\}, 5 \equiv \{3,1,2\}, 6 \equiv \{3,2,1\}\} \tag{2.31}$$

At $t = 0$, the initial configuration is taken to be

$$\pi^{\{t=0\}} = 1 = \delta[(1,,\ldots,n)]. \tag{2.32}$$

This configuration has nothing special to it—it has no lower energy, no higher probability. An appropriate relabeling of cards would map it onto any other configuration. We now take the top card and insert it into the deck in a way described in the transition matrix given by

$$P_n^{\text{shuffle}} = \frac{1}{3}\begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, \tag{2.33}$$

and written up in Alg. 2.10 (`top-to-random`). Clearly, the top-to-random shuffle

procedure `top-to-random`
input $\{c_1, \ldots, c_n\}$
$i \leftarrow$ `choice`$(\{1, \ldots, n\})$
$\{\hat{c}_1, \ldots, \hat{c}_n\} \leftarrow \{c_2, \ldots, c_i, c_1, c_{i+1}, \ldots, c_n\}$
output $\{\hat{c}_1, \ldots, \hat{c}_n\}$

Algorithm 2.10: `top-to-random` Markov-chain Monte Carlo shuffling algorithm for n cards in a deck. A single iteration is shown.

procedure `top-to-random-stop`
input $\{c_1, \ldots, c_n\}$
$c_{\text{first}} \leftarrow c_n$
for $t = 1, 2, \ldots$**:**
  $\tilde{c}_1 \leftarrow c_1$
  $\{\hat{c}_1, \ldots, \hat{c}_n\} \leftarrow$ `top-to-random`$(\{c_1, \ldots, c_n\})$
  if $\tilde{c}_1 = c_{\text{first}}$**: break**
output $\{\hat{c}_1, \ldots, \hat{c}_n\}, t$

Algorithm 2.11: `top-to-random-stop`. Markov-chain Monte Carlo shuffling algorithm for n cards in a deck integrating a stopping rule. It outputs a perfect sample.

Eigenvalues of P_n^{shuffle}: $0, \frac{1}{n}, \frac{2}{n}, \ldots, 1 - \frac{2}{n}, 1$ Degeneracies:

$$\begin{aligned} n = 2 &: [1, 0, 1] \\ n = 3 &: [2, 3, 0, 1] \\ n = 4 &: [9, 8, 6, 0, 1] \\ n = 5 &: [44, 45, 20, 10, 0, 1] \\ n = 6 &: [265, 264, 135, 40, 15, 0, 1] \\ n = 7 &: [1854, 1855, 924, 315, 70, 21, 0, 1] \end{aligned}$$

Although the inverse gap is of order $\mathcal{O}(n)$, we find that there is an intermediate, *non-asymptotic* time scale $\mathcal{O}(n \log n)$ called the mixing time. It was discovered in the example of the top-to-random shuffle.

Expected running time: $n \log n$

- Total variation distance:

$$||\pi^{\{t\}} - \pi||_{\rm TV} = \max_{A\subset\Omega} |\pi^{\{t\}}(A) - \pi(A)| = \frac{1}{2}\sum_{i\in\Omega} |\pi_i^{\{t\}} - \pi_i|.$$

- Distance:

$$d(t) = \max_{\pi^{\{0\}}} ||\pi^{\{t\}}(\pi^{\{0\}}) - \pi||_{\rm TV}$$

- Mixing time:

$$t_{\rm mix}(\epsilon) = \min\{t : d(t) \le \epsilon\}$$

- Usually $\epsilon = 1/4$ is taken, $\epsilon = 1/e$ would be better.
- Theorems: $d[\ell t_{\rm mix}(1/2)] < 2^{-\ell} t_{\rm mix}(1/2)$
- Theorems: $d[\ell t_{\rm mix}(1/e)] < e^{-\ell} t_{\rm mix}(1/e)$

Relaxation time:

- $t_{\rm mix} = ||\pi^{\{t_{\rm mix}\}} - \pi||_{\rm TV} = 1/e$
- $t_{\rm corr}$ = inverse gap.
- $t_{\rm mix} \gg t_{\rm corr}$ leads to cutoff phenomenon.
- Aldous–Diaconis (1986)
- Diaconis–Fill-Pitman (1992)

2.4.3 Perfect sampling

Another approach to sampling exactly ("perfectly") from a distribution π using a Markov chain that starts at $\pi^{\{t=0\}}$.

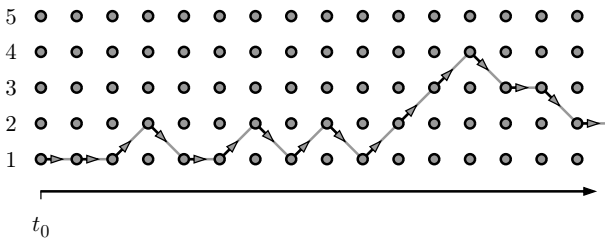

Figure 2.12: Diffusion of a particle on 5 sites

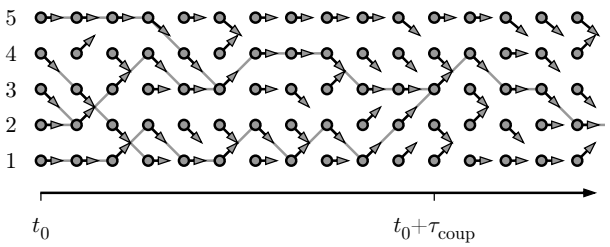

Figure 2.13: Diffusion of a particle on 5 sites using random maps. The position where particles couple is not a sample of π.

procedure `forward-coupling`
$\mathcal{P} \leftarrow \{1, \ldots, N\}$
$t \leftarrow 0$
while True:
$\left\{ \begin{array}{l} t \leftarrow t+1 \\ \mathcal{P} \leftarrow \{\min[\max(b + \texttt{choice}(-1,+1), N] \text{for } b \in \mathcal{P}\} \\ \textbf{if } |\mathcal{P}| = 1\textbf{: break} \end{array} \right.$
output $\mathcal{P}, t$ (position, time of coupling)

Algorithm 2.12: `forward-coupling`. Forward coupling algorithm

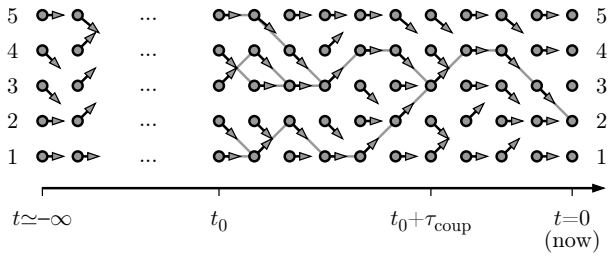

Figure 2.14: Diffusion of a particle on 5 sites using the coupling-from-the-past approach. The position at $t = 0$ is a perfect sample, if we can find it from $t = -\infty$.

References

[1] L. Wasserman, *All of Statistics*. New York: Springer, 2004.

[2] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[3] G. Tartero and W. Krauth, "Concepts in Monte Carlo sampling," *American Journal of Physics*, vol. 92, no. 1, p. 65–77, 2024.

[4] W. Krauth, *Statistical Mechanics: Algorithms and Computations*. Oxford University Press, 2006.

[5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[6] W. Krauth, "Event-Chain Monte Carlo: Foundations, Applications, and Prospects," *Front. Phys.*, vol. 9, p. 229, 2021.

[7] E. A. J. F. Peters and G. de With, "Rejection-free Monte Carlo sampling for general potentials," *Phys. Rev. E*, vol. 85, p. 026703, 2012.

procedure `coupling-from-past`
$t_{\rm tot} \leftarrow 0$
while True:
$\left\{ \begin{array}{l} t_{\rm tot} \leftarrow t_{\rm tot} - 1 \\ \mathcal{A}_{t_{\rm tot}} \leftarrow \texttt{draw-arrows} \text{ (draw arrows at time } t_{\rm tot}) \\ \textbf{for } t = t_{\rm tot}, t_{\rm tot}+1, \ldots, -1\textbf{:} \\ \quad \{ \ \mathcal{P} \leftarrow \{b + \mathcal{A}_t(b) \text{for } b \in \mathcal{P}\} \\ \textbf{if } |\mathcal{P}| = 1\textbf{: break} \end{array} \right.$
output $\mathcal{P}$ (perfect sample)

Algorithm 2.13: `coupling-from-past`. Coupling-from-the-past algorithm for diffusion.